Сучасні каркаси web-додатків

Складені вкладки у Vue

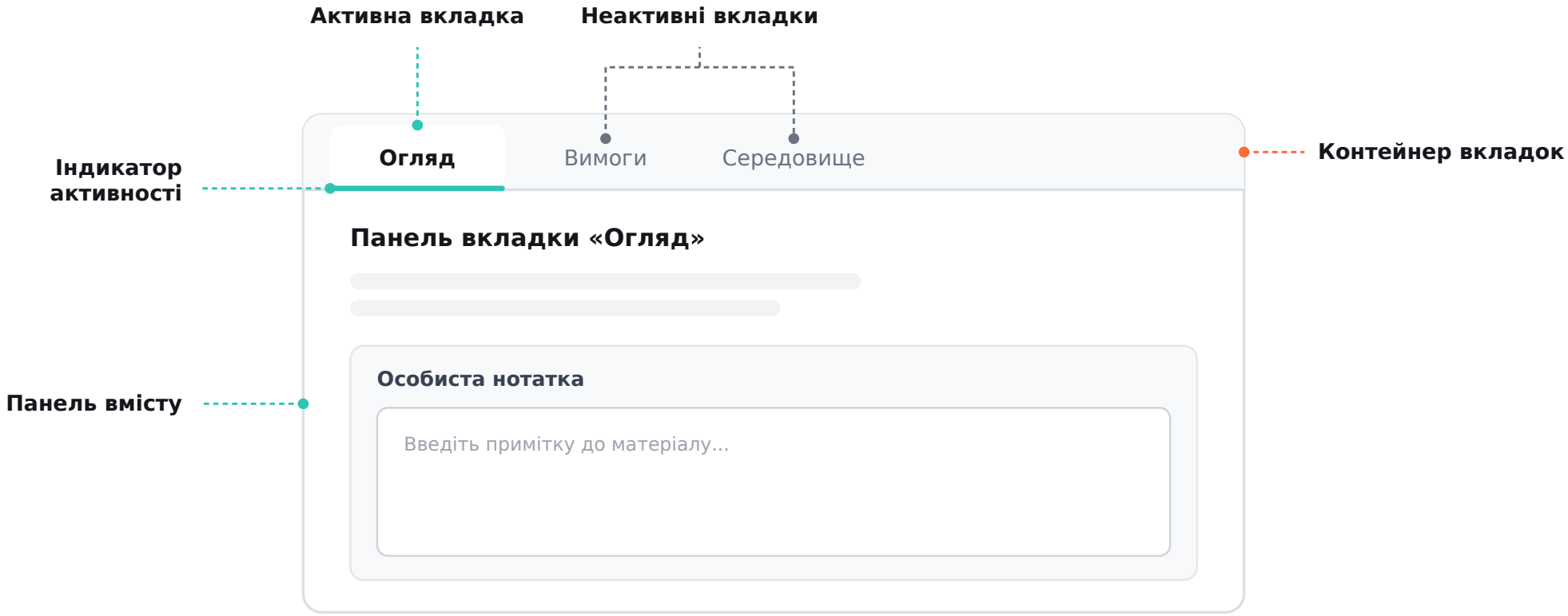
Теорія лабораторної роботи 4

Лабораторна робота 4 · Лісняк Андрій Олександрович

Складений компонент

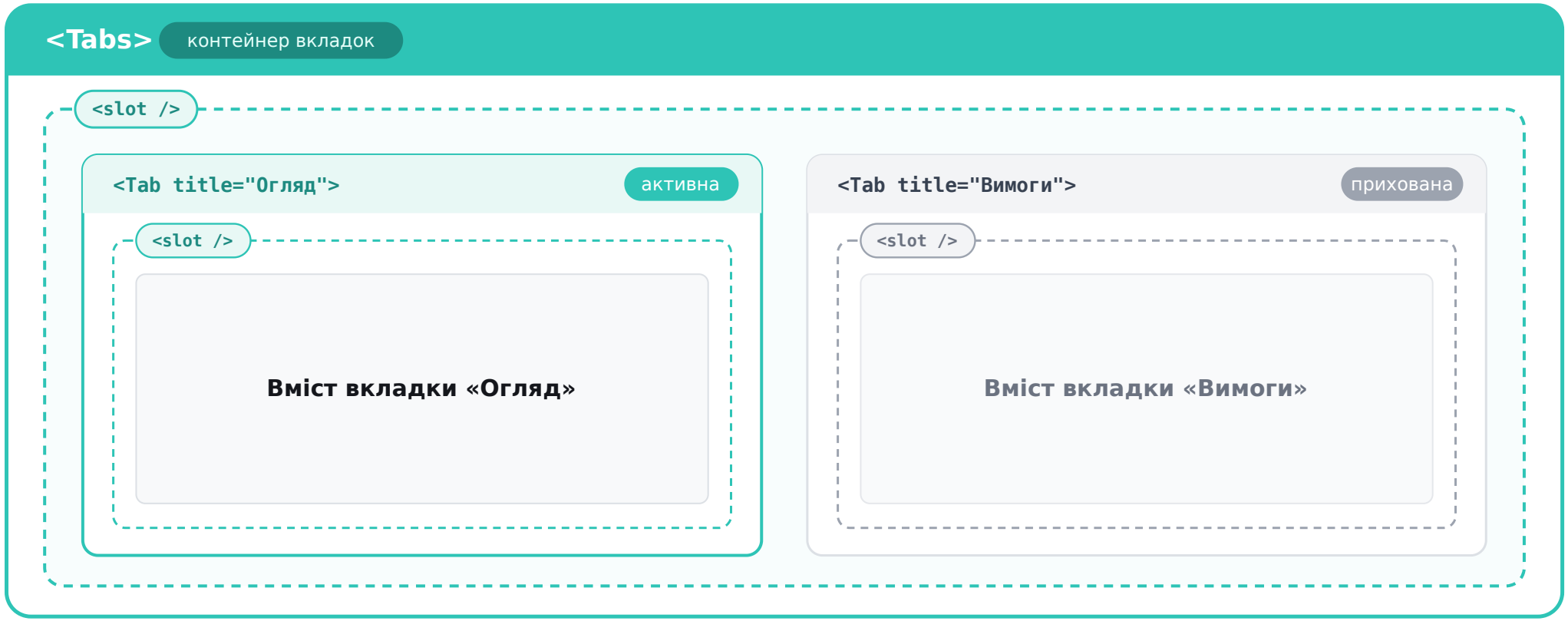
Поняття	Зміст
Проблема	Монолітний компонент накопичує налаштування різних частин інтерфейсу.
Рішення	Кілька компонентів координують поведінку через спільний стан.
Приклади	Система вкладок або багатоступенева форма.
Композиція	Батьківський шаблон визначає склад і порядок частин.

Будова системи вкладок



Смуга кнопок керує показом панелей. У кожній панелі може бути власний стан.

Контейнер Tabs і декларації Tab



Tabs координує набір. Кожен Tab описує одну вкладку та її панель.

Декларативний опис вкладок

src/CompositionDemo.vue · рядки 9–17

```
<template>
  <Tabs v-model="activeTab" variant="underline">
    <Tab slug="overview" title="Огляд">
      <StatePanel label="Нотатка"/>
    </Tab>
    <Tab slug="requirements" title="Вимоги"
      :panel="StatePanel" :panel-props="{ label: 'Вимоги' }"/>
  </Tabs>
</template>
```

`npm ci`

Складені вкладки у Vue

Вміст і параметри слота

Поняття	Зміст
Default slot у Tabs	Містить декларації Tab з батьківського шаблону.
Default slot у Tab	Містить розмітку або компонент конкретної панелі.
Область видимості	Вміст слота використовує змінні того, хто його передає.
isActive	Пропс слота повідомляє панелі, чи вибрана її вкладка.

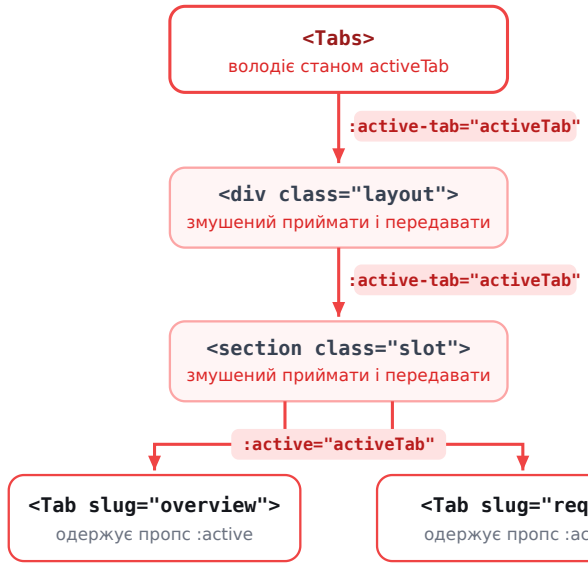
Навіщо потрібен спільний контекст

Поняття	Зміст
Tabs	Має побудувати кнопки для оголошених вкладок.
Tab	Має визначити, чи збігається його slug з активним.
Prop drilling	Проміжні компоненти приймають і передають чужі пропси.
provide / inject	Предок надає контекст нащадкам у своєму піддереві.

Пропси та provide / inject

Проблемний підхід

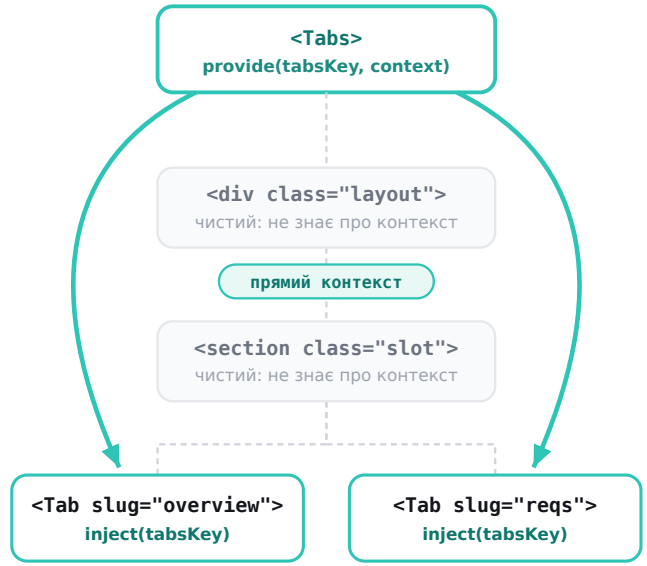
Прокидання пропсів (Prop Drilling)



Недоліки прокидання пропсів:
· Проміжні шари перевантажені чужими параметрами
· Будь-яка зміна структури дерева руйнує передавання даних

Еталонне рішення

Впровадження залежностей (Provide / Inject)



Переваги Provide / Inject:
· Пряма доставка даних крізь будь-яку кількість шарів
· Проміжні компоненти залишаються чистими й незалежними

> Контекст доступний нащадку без передавання через кожний проміжний компонент.

Спільний ключ ін’єкції

src/tabsKey.js

```
export const tabsKey = Symbol('tabs')
```

`npm ci`

Складені вкладки у Vue

Реєстрація декларацій

1. **Монтування** Tab викликає registerTab у своєму onMounted.
2. **Навігація** Tabs будує кнопки за реактивним реєстром.
3. **Вилучення** onBeforeUnmount викликає unregisterTab.
4. **Узгодження** Після вилучення активної вкладки обирається доступна.

Незалежність наборів вкладок

Поняття	Зміст
Кожен Tabs	Виконує власний provide зі своїм станом і реєстром.
Кожен Tab	Отримує контекст найближчого відповідного предка.
Спільний Symbol	Позначає тип зв’язку, не об’єднує стани.
Результат	Перемикання одного набору не змінює інші.

Панель розміткою або значенням

Поняття	Зміст
Слот	Склад відомий у шаблоні; панель і її пропси записані поруч.
panel / panelProps	Склад приходить масивом; тип панелі передається значенням.
Компроміс	У другому випадку параметри панелі проходять через Tab.
Пріоритет	Якщо слот надано, Tab використовує його.

Динамічний компонент і його пропси

src/Tab.vue · рядки 15–24

```
<template>
  <section v-show="isActive" class="tabs__panel">
    <KeepAlive>
      <SlotPanel v-if="isActive && $slots.default">
        <slot :is-active="isActive"/>
      </SlotPanel>
      <component v-else-if="isActive" :is="panel" v-bind="panelProps"/>
    </KeepAlive>
  </section>
</template>
```

`npm ci`

Складені вкладки у Vue

Локальний стан належить екземпляру

src/StatePanel.vue · рядки 1–11

```
<script setup>
import { ref, onMounted, onUnmounted,
  onActivated, onDeactivated } from 'vue'
const props = defineProps({ label: String })
const emit = defineEmits(['event'])
const text = ref('')
const count = ref(0)
onMounted(() => emit('event', props.label + ': mounted'))
onUnmounted(() => emit('event', props.label + ': unmounted'))
onActivated(() => emit('event', props.label + ': activated'))
onDeactivated(() => emit('event', props.label + ': deactivated'))
```

`npm ci`

Складені вкладки у Vue

DOM і стан неактивної панелі

Поняття	Зміст
v-show	DOM залишається; поточні значення ref зберігаються.
v-if без кешу	Екземпляр демонтується; повернення створює початковий стан.
KeepAlive + умова	Панель вилучається з видимого DOM; екземпляр зберігається.

Життєвий цикл і вибір підходу

Поняття	Зміст
v-show	Без деактивації; підходить для кількох простих панелей.
v-if	Unmount і новий mount; стан можна починати заново.
KeepAlive	Deactivate й activate; потрібний для форм та фільтрів.

Приховування через v-show і v-if

src/ModeDemo.vue · рядки 13–17

```
    <button @click="open = !open">Перемкнути панель</button>
    <StatePanel v-if="mode === 'v-show'" v-show="open"
      label="Панель" @event="record"/>
    <StatePanel v-else-if="mode === 'v-if' && open"
      label="Панель" @event="record"/>
```

`npm ci`

Складені вкладки у Vue

Панель усередині KeepAlive

src/ModeDemo.vue · рядки 18–20

```
    <KeepAlive v-else-if="mode === 'KeepAlive'">
      <StatePanel v-if="open" label="Панель" @event="record"/>
    </KeepAlive>
```

`npm ci`

Складені вкладки у Vue

Життя кешованої панелі

1. **Перше відкриття** setup, mounted, activated; значення початкові.
2. **Редагування** Користувач змінює локальні ref панелі.
3. **Інша вкладка** deactivated; екземпляр перебуває в кеші.
4. **Повернення** activated; попередні значення зберігаються.

Хуки кешованого компонента

Поняття	Зміст
onActivated	Перше монтування та кожне повернення з кешу.
onDeactivated	Вихід у кеш і остаточний демонтаж.
Фонові процеси	Таймери та слухачі потребують явної зупинки.
Очищення	Повторні активації не повинні дублювати ресурси.

Межі збереження стану

Поняття	Зміст
Зберігається	Екземпляр компонента з його ref та локальними даними.
Повернення	Той самий тип і стабільний ключ знаходять той самий екземпляр.
Втрата кешу	Демонтаж KeepAlive або витіснення завершує життя екземпляра.
Оновлення сторінки	Починається новий запуск застосунку з початковими даними.

Обмеження кешу через max

src/CacheDemo.vue · рядки 13–16

```
    <KeepAlive :max="2">
      <component :is="StatePanel" :key="active"
        :label="active" @event="events.push($event)"/>
    </KeepAlive>
```

`npm ci`

Складені вкладки у Vue

LRU-кеш із межею два

1. **A, потім B** У кеші два екземпляри, A використовувався раніше.
2. **Знову A** A стає останнім використаним.
3. **Відкрити C** Кеш витісняє B і знищує його стан.
4. **Повернути B** Створюється новий екземпляр B з початковими даними.

KeepAlive у компоненті Tab

Поняття	Зміст
Декларація Tab	Залишається змонтованою, поки присутня у складі набору.
Вміст панелі	Входить до кешу, коли isActive стає false.
Вміст слота	SlotPanel утворює екземпляр, дерево якого кешується.
Власний кеш	Кожен Tab у цьому шаблоні має окремий KeepAlive.

Перевірка збереження стану

1. **Змінити** Ввести нотатку і кілька разів збільшити лічильник.
2. **Перемкнути** Відкрити іншу панель, перевірити журнал подій.
3. **Повернути** Зіставити значення до виходу та після повернення.
4. **Повторити** Перевірити слот і panelProps, а потім оновити сторінку.

Чотири інтерфейси IDEF0

Поняття	Зміст
Input, ліворуч	Дані, які функція опрацьовує.
Control, згори	Правила та обмеження виконання.
Output, праворуч	Результат виконання функції.
Mechanism, знизу	Засоби, які забезпечують виконання.

A1. Керувати вкладками



> Tabs отримує декларації й команди, формує навігацію та контекст.

A2. Показати панель



Tab реєструє вкладку та відображає її вміст за активністю з контексту.

Ролі WAI-ARIA для вкладок

Поняття	Зміст
tablist	Список кнопок із доступною назвою.
tab	Кнопка вибору пов’язаної панелі.
tabpanel	Контейнер вмісту конкретної вкладки.

Назви, зв’язки та активність

Поняття	Зміст
Список	aria-label або aria-labelledby задає назву.
Кнопка	aria-controls вказує на id її панелі.
Панель	aria-labelledby вказує на id її кнопки.
Вибір	aria-selected=true лише у вибраної вкладки.

Roving tabindex

Поняття	Зміст
Активна кнопка	Єдина кнопка набору в послідовності Tab.
Інші кнопки	tabindex=-1; фокус отримують програмно.
Вхід	Tab із попереднього елемента веде до активної кнопки.
Панель	tabindex=0 дозволяє перейти до початку її вмісту.

Автоматична активація стрілками

1. **ArrowRight** Наступна вкладка отримує фокус і стає активною.
2. **Правий край** Після останньої вибирається перша.
3. **ArrowLeft** Попередня вкладка отримує фокус і стає активною.
4. **Лівий край** Перед першою вибирається остання.

Home, End і вихід із набору

Поняття	Зміст
Home	Фокус і активація першої доступної вкладки.
End	Фокус і активація останньої доступної вкладки.
Tab із кнопки	Перехід до активної панелі, далі до її елементів.
Shift+Tab	Зворотний рух у порядку фокусованих елементів.

Складені вкладки та стан панелей

Дорожня карта

1. **Композиція** Tabs координує, Tab описує, слот передає вміст.
2. **Контекст** provide/inject ізолює кожен набір.
3. **Кеш** KeepAlive зберігає локальний екземпляр.

Завдання

- Пояснити різницю v-show, v-if і KeepAlive
- Показати збереження стану обома способами
- Перевірити клавіатурну навігацію